Sequence Labeling

Seungwoo Schin

Natural Language Paper Reading Group, Tensorflow KR

In this session, we will dive into two possible ways for sequence tagging. First, we will deal with theory of sequence tagging and establish two models. Than we will implement the model, and test each model's performance on POS tagging and NER problem. Codes will be given with this document.

I. Background

I. Sequence Tagging

Formal definition of sequence labeling would be as below; for a given sequence of tokens $\{x_i\}$, we want best guess for each x. We denote possible options for tags as y_i, and we assume that each label is a integer between 1 and L.

As most classification tasks does, we take discriminative model[1]; that is, to analyze the model according to the given test input. So we assume scoring function $s(y, x)$ here, and assume that the best prediction for a given sequence x is equivalent to solve following inference problem;

$$\hat{y} = argmax_y s(\vec{y}, \vec{x}) \tag{1}$$

So now the problem is reduced to learn and evaluate the scoring function. In today's session, we will try two methods: logistic regression model and conditional random field(CRF) model.

I.1 Logistic Regression

For the base case, we assume complete independence between each component in the resulting sequence. Thus, scoring function $s(\vec{y}, \vec{x})$ can be decomposed to each terms, as following;

$$s(\vec{y}, \vec{x}) = \sum \psi(y_i, i, \vec{x}) \tag{2}$$

and now the problem reduces to find most suitable ψ function. Now, we want to solve this problem by assigning features on each token[2]. Since we are now dealing with discrete labels, it is possible to consider only binary features and apply logistic regression.

Using aforementioned settings, logistic regression can be written as following;

$$\psi(y_i, i, x) = \theta_y \phi(i, x) \tag{3}$$

where $\phi(i, x)$ is a feature vector function. So, let's assume that there are f features and L labels. Than ϕ will generate $f \times 1$ vector, and θ_y will be $L \times f$ matrix. Now solving argmax problem is easy when θ_y is once obtained; we can simply choose the label that gives maximum value for each x_i, and greedy algorithm is enough to do so.

[1]If any one of you have knowledge about generative model, please tell me during the presentation.

[2]To be precise, token itself does not serve as a complete input for feature generation. Note that the scoring funciton is given as $\psi(y_i, i, \vec{x})$, not $\psi(y_i, x_i)$. Whole sequence might be used for feature; for instance, if we want to generate a feature that indicates whether the token is the first word of the sentence, we will need to have whole sequence for extracting that token.

So how do we get θ_y? We simply do logistic regression for f times on each row. So each testcase will cover each row, and since there are f rows, we can do it f times and that will generate L coefficients as weight, and that will do the trick.

I.2 Conditional Random Field

Now we move on to slightly more general setting; we will assume Markovian Property this time. Therefore, score function will be now as below;

$$s(\vec{y}, \vec{x}) = \sum \psi_t(y_{i-1}, y_i) + \psi_s(y_1) + \psi_e(y_n) + \sum \psi(y_i, i, \vec{x}) \tag{4}$$

Settings on dimensions are almost similar to the above version. For learning weights, we will use structured perceptron[3]. For the structured perceptron, we will just use it as a black-box, since today's main topic is how to inference after all weights are obtained. For the first case, we can easily do with greedy approach. However, for this case, greedy algorithm will not work, and brute force algorithm will result in exponential time. Therefore, we apply dynamic programming for choosing the right label. This specific algorithm is called **viterbi algorithm**, and it is so widely used in inference of optimal sequence with Markovian Property. Vieterbi algorithm and its evaluation code is implemented on viterbi.py and viterbi_test.py file.

Core part of implemenation is as following;

```
T1 = []
T2 = []

# after initialization...

for j in range(1, N):
        for i in range(L):
                max_val = -np.inf
                for elem in range(L):
                        tmp_path = T2[elem][j-1] + [i]
                        tmp_val = get_score(tmp_path)
                        if tmp_vall > max_val:
                                # do update...

return argmax(T[i][-1])
```

II. Implementation Details

I. Feature Generation

Four kinds of features are generated for the custom feature. Simple features would be boolean denoitn whether the first letter of the word is capitalized, while the token is not the starting token of the sentence. Also, length of token is checked whether the token have more than 7 charecters.

More complex-and thus interesting-features make use of external data sources. First, for POS tagging task, wordnet is utilized. Since wordnet provides all possible candidates of synonym sets (denoted synsets), it is possible to check which pos does each token most frequently used. Since wordnet provides 5 POS tags(noun, verb, adjective, adjective sat, adverb), 5 features are added for

[3]Ported from https://github.com/pystruct/pystruct/blob/master/pystruct/learners/structured_perceptron.py

denoting each pos. Other feature is generated by crawling external data. US citizen name from cencus data is crawled and saved in a file, and if token is in the list, the feature 'name' is marked true. This feature is intended to play a contribution in NER task. Even though there are massive open data for list of proper nouns, lack of computer memory and time makes it impossible to adapt ontological corpus such as freebase. However, implemented code can be easily scaled to such corpus.

III. Result & Analysis

The result is as following;

I. NER, logistic model

```
NER, Logistic, without additional feature

Token-wise accuracy 93.4267317625
Token-wise F1 (macro) 71.8060689149
Token-wise F1 (micro) 93.4267317625
Sentence-wise accuracy 57.5
           precision   recall f1-score  support

     I-LOC      0.77     0.73     0.75      628
    I-MISC      0.79     0.43     0.55      283
     I-ORG      0.76     0.40     0.53      536
     I-PER      0.73     0.84     0.78     1097
         O      0.97     0.99     0.98    11589

avg / total     0.93     0.93     0.93    14133

NER, Logistic, with additional feature

Token-wise accuracy 94.2262789217
Token-wise F1 (macro) 74.3021648303
Token-wise F1 (micro) 94.2262789217
Sentence-wise accuracy 58.8
           precision   recall f1-score  support

     I-LOC      0.74     0.73     0.74      628
    I-MISC      0.77     0.48     0.59      283
     I-ORG      0.70     0.46     0.56      536
     I-PER      0.81     0.88     0.84     1097
         O      0.98     0.99     0.98    11589

avg / total     0.94     0.94     0.94    14133
```

II. NER, CRF model

```
NER, CRF, without additional feature
```

```
Token-wise accuracy 93.5823958112
Token-wise F1 (macro) 72.9815772114
Token-wise F1 (micro) 93.5823958112
Sentence-wise accuracy 58.6
            precision   recall  f1-score   support

      I-LOC      0.82     0.74      0.78       628
     I-MISC      0.65     0.50      0.57       283
      I-ORG      0.75     0.40      0.52       536
      I-PER      0.77     0.84      0.80      1097
          O      0.97     0.99      0.98     11589

avg / total      0.93     0.94      0.93     14133

NER, CRF, with additional feature

Token-wise accuracy 93.603622727
Token-wise F1 (macro) 73.4569460332
Token-wise F1 (micro) 93.603622727
Sentence-wise accuracy 50.7
            precision   recall  f1-score   support

      I-LOC      0.73     0.74      0.74       628
     I-MISC      0.65     0.60      0.63       283
      I-ORG      0.74     0.39      0.51       536
      I-PER      0.77     0.87      0.82      1097
          O      0.98     0.99      0.98     11589

avg / total      0.93     0.94      0.93     14133
```

III. POS, logistic model

```
POS, Logistic, without additional feature

Token-wise accuracy 84.0019811788
Token-wise F1 (macro) 60.5694566874
Token-wise F1 (micro) 84.0019811788
Sentence-wise accuracy 31.4
            precision   recall  f1-score   support

          "      1.00     1.00      1.00       156
          $      1.00     0.96      0.98        28
         ''      0.00     0.00      0.00         3
          (      1.00     1.00      1.00       300
          )      1.00     1.00      1.00       301
          ,      1.00     1.00      1.00       558
          .      1.00     1.00      1.00       488
          :      1.00     0.97      0.98       143
         CC      1.00     0.95      0.97       219
         CD       0.87      0.96      0.91      1283
         DT       1.00      0.96      0.98       837
         EX       1.00      0.33      0.50         9
         FW       0.00      0.00      0.00         6
         IN       0.92      0.95      0.93      1274
         JJ       0.66      0.44      0.53       773
        JJR       0.00      0.00      0.00        31
        JJS       0.00      0.00      0.00        16
         MD       1.00      0.94      0.97        67
         NN       0.58      0.86      0.69      1596
        NNP       0.85      0.98      0.91      2682
       NNPS       0.00      0.00      0.00        31
        NNS       0.83      0.42      0.56       671
        PDT       0.00      0.00      0.00         1
        POS       0.91      1.00      0.95       145
        PRP       1.00      0.85      0.92       220
       PRP$       1.00      0.87      0.93       103
         RB       0.86      0.38      0.52       289
        RBR       1.00      0.06      0.12        16
        RBS       0.00      0.00      0.00         6
         RP       0.83      0.21      0.34        47
        SYM       1.00      0.38      0.55        40
         TO       1.00      0.98      0.99       215
         UH       0.00      0.00      0.00         2
         VB       0.88      0.76      0.81       274
        VBD       0.82      0.75      0.78       565
        VBG       0.92      0.26      0.40       182
        VBN       0.66      0.68      0.67       238
        VBP       0.77      0.41      0.54        82
        VBZ       1.00      0.48      0.64       141
        WDT       1.00      0.71      0.83        34
         WP       1.00      0.85      0.92        40
        WP$       0.00      0.00      0.00         4
        WRB       1.00      0.12      0.21        17

avg / total       0.85      0.84      0.83     14133

POS, Logistic, with additional feature

Token-wise accuracy 86.9312955494
Token-wise F1 (macro) 65.811682556
Token-wise F1 (micro) 86.9312955494
Sentence-wise accuracy 30.9
             precision    recall  f1-score   support

          "       1.00      1.00      1.00       156
          $       1.00      0.96      0.98        28
         ''       0.00      0.00      0.00         3
          (       1.00      1.00      1.00       300
          )       1.00      1.00      1.00       301
          ,       1.00      1.00      1.00       558
          .       1.00      1.00      1.00       488
          :       1.00      0.96      0.98       143
         CC       1.00      0.95      0.98       219
         CD       0.93      0.93      0.93      1283
         DT       0.99      0.96      0.98       837
         EX       1.00      0.44      0.62         9
         FW       0.00      0.00      0.00         6
         IN       0.92      0.96      0.94      1274
         JJ       0.72      0.69      0.71       773
        JJR       0.00      0.00      0.00        31
        JJS       0.00      0.00      0.00        16
         MD       1.00      0.93      0.96        67
         NN       0.72      0.85      0.78      1596
        NNP       0.85      0.99      0.91      2682
       NNPS       0.00      0.00      0.00        31
        NNS       0.87      0.42      0.57       671
        PDT       0.00      0.00      0.00         1
        POS       0.91      1.00      0.95       145
        PRP       1.00      0.84      0.91       220
       PRP$       1.00      0.89      0.94       103
         RB       0.85      0.73      0.78       289
        RBR       0.50      0.06      0.11        16
        RBS       1.00      1.00      1.00         6
         RP       0.72      0.28      0.40        47
        SYM       1.00      0.38      0.55        40
         TO       1.00      0.98      0.99       215
         UH       0.00      0.00      0.00         2
         VB       0.90      0.78      0.84       274
        VBD       0.73      0.91      0.81       565
        VBG       0.76      0.40      0.52       182
        VBN       0.61      0.69      0.65       238
        VBP       0.85      0.40      0.55        82
        VBZ       0.99      0.48      0.65       141
        WDT       0.96      0.76      0.85        34
         WP       1.00      0.80      0.89        40
        WP$       0.00      0.00      0.00         4
        WRB       1.00      0.41      0.58        17

avg / total       0.87      0.87      0.86     14133
```

IV. POS, CRF model

```
POS, CRF, without additional feature
```

```
POS, CRF, with additional feature
```